

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-044292

(43)Date of publication of application : 14.02.2003

(51)Int.Cl.

G06F 9/46

(21)Application number : 2001-234611

(71)Applicant : KYUSHU UNIV

(22)Date of filing : 02.08.2001

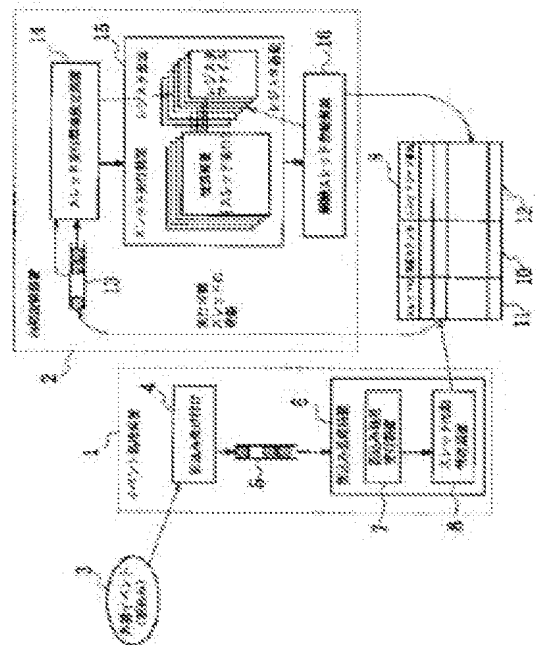
(72)Inventor : AMAMIYA MASATO  
TANIGUCHI HIDEO

## (54) MULTIPLEX PARALLEL PROCESSOR, MULTIPLEX PARALLEL PROCESSING METHOD AND MULTIPLEX PARALLEL PROCESSING PROGRAM EXECUTED BY COMPUTER

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To reduce overheads at switching a program execution environment between the execution of a program and an interruption without complicating the constitution of a computer and basic software and without remarkably increasing the burdens of the description of the program of a user.

**SOLUTION:** An internal arithmetic unit 2 executes the program divided into a plurality of threads by a thread unit. An event processor 1 holds the interruption from the outside while executing a prescribed thread among the threads and executes the interruption after the prescribed thread is ended. The threads are automatically divided by a language processing system.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2003-44292  
(P2003-44292A)

(43) 公開日 平成15年2月14日 (2003.2.14)

(51) Int.Cl.<sup>7</sup>

G 0 6 F 9/46

識別記号

3 1 1

3 4 0

F I

G 0 6 F 9/46

ターマコード\* (参考)

3 1 1 E 5 B 0 9 8

3 4 0 B

審査請求 有 請求項の数 3 O L (全 5 頁)

(21) 出願番号 特願2001-234611(P2001-234611)

(22) 出願日 平成13年8月2日(2001.8.2)

(71) 出願人 391012501

九州大学長

福岡県福岡市東区箱崎6丁目10番1号

(72) 発明者 雨宮 真人

福岡県福岡市南区2-15-10

(72) 発明者 谷口 秀夫

福岡県福岡市早良区小田部3-3-33

(74) 代理人 100072051

弁理士 杉村 興作 (外1名)

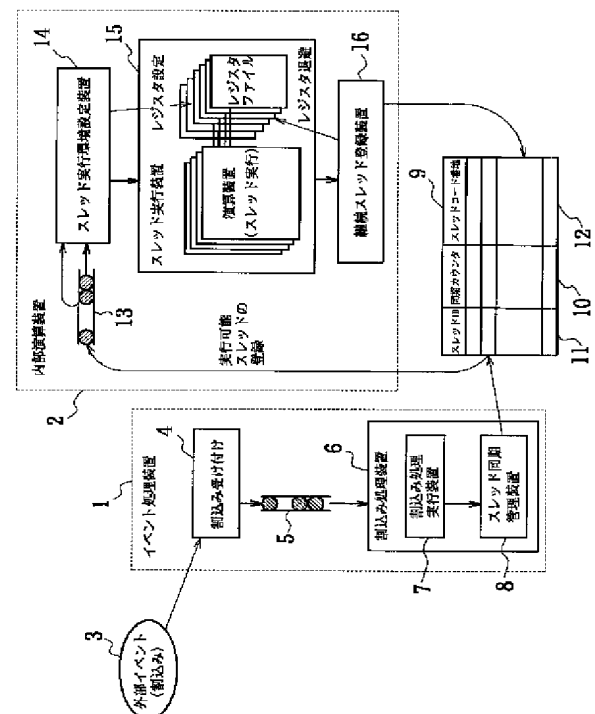
Fターム(参考) 5B098 BA14 BB14 BB15 BB16 GA02  
GA05

(54) 【発明の名称】 多重並行処理装置、多重並行処理方法及びコンピュータによって実行される多重並行処理プログラム

(57) 【要約】

【課題】 計算機及び基本ソフトウェアの構成を複雑にすることなく、かつ、ユーザのプログラムの記述の負担を著しく大きくすることなく、プログラムの実行と割込みとの間のプログラム実行環境の切替の際のオーバーヘッドを軽減する。

【解決手段】 内部演算装置2は、複数のスレッドに分割されたプログラムをスレッド単位に実行する。イベント処理装置1は、これら複数のスレッドのうちの所定のスレッドの実行中に外部からの割込みを保持し、所定のスレッドの終了後、割込みを実行する。複数のスレッドは、言語処理系によって自動的に分割される。



**【特許請求の範囲】**

**【請求項1】** 既存のプログラムを言語処理系によって複数のスレッドプログラムに分割する手段と、そのスレッドプログラムの実行中に外部からの割り込みを保持する手段と、

前記スレッドプログラムの終了後、前記割り込みを実行する手段とを具えることを特徴とする多重並行処理装置。

**【請求項2】** 既存のプログラムを言語処理系によって複数のスレッドプログラムに分割するステップと、そのスレッドプログラムの実行中に外部からの割り込みを保持するステップと、

前記スレッドプログラムの終了後、前記割り込みを実行するステップとを具えることを特徴とする多重並行処理方法。

**【請求項3】** 既存のプログラムを言語処理系によって複数のスレッドプログラムに分割するステップと、そのスレッドプログラムの実行中に外部からの割り込みを保持するステップと、

前記スレッドプログラムの終了後、前記割り込みを実行するステップとを具えることを特徴とするコンピュータによって実行される多重並行処理プログラム。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】** 本発明は、ネットワークに結合した各種の並列及び分散計算機で利用され、マルチスレッド処理の原理を適用して、従来のプロセスをスレッドと呼ぶ排他的に実行可能なプログラム断片に細分化し、それを多重化して並行に走行させることによって、通信や入出力のような外部イベントの処理と計算機内部の処理とを統合し、その並列実行を制御する多重並行処理装置、多重並行処理方法及びコンピュータによって実行される多重並行処理プログラムに関するものである。

**【0002】**

**【従来の技術】** 従来のイベント処理では、外部からの割り込み（イベント）が発生する度に、現在実行中のプログラムを中断し、その割り込みを優先的に処理している。この場合、割り込みの度に、通常「コンテキスト切替」と呼ばれるプロセッサの実行環境の切替を行い、そのイベントの到着を待った後に、実行を中断しているプログラムの実行を再開しているので、オーバーヘッドが増大する。

**【0003】** また、従来のイベント処理において、イベントの到着を待って実行を中断しているプログラムと該当イベント到着とのタイミングを正確にとるようにより制御するプログラムを設計する必要がある。特に、並列及び分散処理では、複数の計算機が互いにデータの授受を行いながら同時並行的に動作しているので、他のプロセッサとの通信やデータの授受に起因するイベントが頻発する。そのようなイベントによって、各プロセッサ上でのプログラム実行順序の制御すなわち同期制御の機構が複雑となるため、プログラム処理を単純化することが所望

されている。

**【0004】**

**【発明が解決しようとする課題】** イベントを到達直後に取り上げて処理する従来のイベント処理では、計算機及び基本ソフトウェアの構成並びに並列及び分散処理のプログラム構成の際に、以下の課題がある。

**【0005】** (1) 従来の計算機ハードウェア及びオペレーティングシステム(OS)においては、主に、コンテキスト切替のオーバーヘッドをできるだけ小さくすることを目的としている。したがって、ユーザプログラムを中心とする内部演算処理と外部要因で発生するイベント処理とを区別し、その切替に着目した制御を行う必要がある。これによって、計算機及び基本ソフトウェアの構成が複雑になり、開発コストが増大する。

**【0006】** (2) プログラム記述、特に、並列及び分散処理のプログラム記述において、イベント待ちとイベント到着のタイミングを予測し、タイミングのミスマッチをできるだけ小さくし、かつ、コンテキスト切替の回数をできるだけ少なくするようにプログラムを設計し及び記述することが要求されている。並列及び分散処理を設計し及び記述するユーザにとっては、外部発生イベントと内部演算処理とを互いに考慮して処理を記述する必要があるため、本来処理すべきアルゴリズムのみに専念してプログラムを記述するのが困難となり、ユーザのプログラム記述の負担が著しく大きくなっている。

**【0007】** 本発明の目的は、計算機及び基本ソフトウェアの構成を複雑にすることなく、かつ、ユーザのプログラムの記述の負担を著しく大きくすることなく、プログラムの実行と割り込みとの間のプログラム実行環境の切替の際のオーバーヘッドを軽減する多重並行処理装置、多重並行処理方法及びコンピュータによって実行される多重並行処理プログラムを提供することである。

**【0008】**

**【課題を解決するための手段】** 本発明による多重並行処理装置は、既存のプログラムを言語処理系によって複数のスレッドプログラムに分割する手段と、そのスレッドプログラムの実行中に外部からの割り込みを保持する手段と、前記スレッドプログラムの終了後、前記割り込みを実行する手段とを具えることを特徴とする。

**【0009】** 本発明によれば、既存のプログラムを言語処理系によって複数のスレッドプログラムに分割し、外部からの割り込みを、スレッドプログラムの終了後に実行するので、割り込みの度にプログラムを中断する必要がなくなる。このように、複数のスレッドに分割されたプログラムをスレッド単位で実行することによって、プログラムの実行と割り込みとの間のプログラム実行環境の切替の際にオーバーヘッドを軽減することができる。なお、本発明では、外部から複数の割り込みがあったとしても、スレッド終了後に複数の割り込みをまとめて行うことができるので、プログラムの実行と割り込みとの間のプログラム

実行環境の切替を行う回数を著しく低減し、その結果、プログラムの実行と割込みとの間のプログラム実行環境の切替のために計算機及び基本ソフトウェアの構成を複雑にする必要がなくなり、かつ、ユーザのプログラムの記述の負担が著しく大きくなるおそれがない。

【0010】本発明による多重並行処理方法は、既存のプログラムを言語処理系によって複数のスレッドプログラムに分割するステップと、そのスレッドプログラムの実行中に外部からの割込みを保持するステップと、前記スレッドプログラムの終了後、前記割込みを実行するステップとを具えることを特徴とする。

【0011】本発明によれば、計算機及び基本ソフトウェアの構成を複雑にすることなく、かつ、ユーザのプログラムの記述の負担を著しく大きくすることなく、プログラムの実行と割込みとの間のプログラム実行環境の切替の際のオーバーヘッドを軽減することができる。

【0012】本発明によるコンピュータによって実行される多重並行処理プログラムは、既存のプログラムを言語処理系によって複数のスレッドプログラムに分割するステップと、そのスレッドプログラムの実行中に外部からの割込みを保持するステップと、前記スレッドプログラムの終了後、前記割込みを実行するステップとを具えることを特徴とする。

【0013】本発明によれば、計算機及び基本ソフトウェアの構成を複雑にすることなく、かつ、ユーザのプログラムの記述の負担を著しく大きくすることなく、プログラムの実行と割込みとの間のプログラム実行環境の切替の際のオーバーヘッドを軽減する多重並行処理をコンピュータプログラムによって実行することができる。

#### 【0014】

【発明の実施の形態】本発明による多重並行処理装置、多重並行処理方法及びコンピュータによって実行される多重並行処理プログラムの実施の形態を、図面を参照して詳細に説明する。

【0015】先ず、第1の実施の形態について説明する。第1の実施の形態では、イベント処理とスレッド実行（内部演算処理）を独立して並列に行う機構をハードウェアで実現する。本実施の形態では、言語処理系によって既存のプログラムコードを図1に示すようなスレッドプログラム構造に変換したプログラムコードを実行することを前提とし、本発明による多重並行処理装置を構成する図2のイベント処理装置1及び内部演算装置2によって処理が実行される。

【0016】図1において、a, b, c, d, eはそれぞれ、スレッドコード、スレッド起動条件（同期カウンタ）、スレッド実行命令列、スレッド実行終了命令及び継続スレッド情報（ポインタ）を示す。

【0017】イベント処理装置1及び内部演算装置2は並列に動作する。イベント処理装置1は、外部イベント（割込み）3の処理及びスレッド同期管理を行い、内部

演算装置2は、実行可能スレッドの実行を専用に行う。

【0018】この状況で外部イベント3が発生すると、それは、イベント処理装置1の割込み受付装置4によって取り上げられ、イベント処理装置1の割込みキュー5に保持される。

【0019】イベント処理装置1の割込み処理装置6は、割込みキュー5に保持されている割込みを1個ずつ取り出して割込み処理を行う。割込み処理装置6は、割込み処理実行装置7及びスレッド同期管理装置8を有し、次の動作を行う。

【0020】1. 割込み要因に基づいて、該当する割込み処理ルーチンを実行する。

2. スレッド同期管理装置8は、同期待ちスレッド管理表9の中に保持されている同期待ちスレッドに対し、割込み処理の結果に従って、その割込みに起因するイベントを待つスレッドの同期カウンタ10を1減じる。なお、割込みに起因するイベントを待つスレッドは、スレッドID11によって識別される。

3. 同期カウンタ10を1減じた結果、その値が零になると、当該スレッドのスレッドID11と、スレッドコード番地12とを、内部演算装置2内の実行待ちキュー13に入れる。

【0021】内部演算装置2は、実行待ちスレッドキュー13に保持されている実行可能キューを取り上げ、スレッドの実行を行う。内部演算装置2は、スレッド実行環境設定装置14、スレッド実行装置15及び継続スレッド登録装置16を更に有し、以下の処理を行う。

【0022】1. スレッド実行環境設定装置14は、実行の順番が近づいてきた、すなわち、キューの先頭近くに移動してきた実行間待ちスレッドのために、予め空きレジスタファイル割り当て、そのレジスタファイルに当該スレッドの実行環境を設定する（実行環境の回復）。

2. 実行すべきスレッドを実行待ちキュー13の先頭から取り出し、現在空き状態の演算装置に対して、既に割り当てられているレジスタファイルを用いてスレッド実行を開始する。スレッドの実行は、途中で中断することなくスレッド実行終了命令の実行まで命令列を排他的に実行する。

3. 継続スレッド登録装置16は、スレッド実行終了命令を実行する。スレッド実行終了命令は、当該スレッドに続いて実行すべきスレッド（継続スレッド）の名前（スレッドID及びそのコード番地）及びその同期カウンタの初期値をオペランドとして持っているため、そのオペランド情報を用いて、継続スレッドを同期待ちスレッド管理表9に登録する。同一の継続スレッドが同期待ちスレッド管理表9に既に存在する場合、その同期カウンタ10を1減じる。

4. 同期カウンタ10を1減じた結果、その値が零になり、起動条件が整ったときには、当該スレッドのスレ

ドID及びコード番地を内部演算装置 2 内の実行待ちキュー 1 3 に入れる。

【0 0 2 3】次に、第 2 の実施の形態を説明する。第 2 の実施の形態は、イベント処理とスレッド実行（内部演算処理）を独立して並列に行う機構をソフトウェアで実現する。本実施の形態でも、言語処理系によって既存のプログラムコードを図 1 に示すようなスレッドプログラム構造に変換したプログラムコードを実行することを前提とし、本発明による多重並行処理方法を実施する図 3 に示すような市販のハードウェア装置 2 1 によって処理が

【0 0 2 4】ハードウェア装置 2 1 は、演算装置 2 2 と、外部装置（割込み装置）2 3 と、記憶装置 2 4 と、これらの間でデータのやり取りを行うバス 2 5 とを有する。

【0 0 2 5】本実施の形態の動作を、図 4 も用いて説明する。演算装置 2 2 でスレッド実行管理部 2 6 が実行されると同時に、外部イベント 2 7 は、外部装置 2 3 によって取り上げられ、記憶装置 2 4 内の割込みキュー 2 7 に保持される。スレッド実行管理部 2 6 は、スレッド実行部 2 9 及び割込み処理実行部 3 0 を有する。ここで、スレッドの実行環境とはレジスタの内容である。

【0 0 2 6】スレッド実行中にスレッド実行終了命令を実行すると、そのスレッドの実行が終了し、スレッド実行環境退避部でスレッド実行環境の退避を行う。スレッド実行終了命令には、当該スレッドに継続する少なくとも一つのスレッドの情報がオペランドとして示されており、この継続スレッドを記憶装置 2 4 の同期待ちスレッド管理表 3 1 に登録する。同一の継続スレッドが既に同期待ちスレッド管理表 3 1 に存在する場合、その同期カウンタを 1 減じる。

【0 0 2 7】スレッド実行部 2 9 は、実行待ちスレッドキュー 3 2 に保持されている実行待ちスレッドをキューの先頭から取り出し、そのスレッドの実行環境をスレッド実行環境設定部で設定し、スレッド実行部 2 9 でスレッドコード（実行命令列）を実行する。

【0 0 2 8】一つのスレッド実行が終了すると、割込み処理実行部 3 0 に制御を移行する。割込み処理実行部 3 0 では、割込みスレッドキュー 2 8 が空になるまで次の操作を繰り返す。

【0 0 2 9】1. 割込みキュー 2 8 の先頭から割込みを取り出し、その割込み要因を調べ、該当する割込み処理ルーチンを実行する。

2. そのイベントを待っている、すなわち、同期待ちスレッド管理表 3 2 内に登録されているスレッドに対して、その同期カウンタを 1 減じる（スレッド同期処理）。

3. その結果として同期カウンタの値が零となった場合には、そのスレッドを、同期待ちスレッド管理表 3 1 から削除し、実行待ちスレッドキュー 3 2 に入れる。割込

み実行部 3 0 の処理が終了すると、制御をスレッド実行部 2 9 に移行する。

【0 0 3 0】本実施の形態によれば、イベント処理と内部演算処理の特性を分析することによって、並列及び分散処理を含む一般プログラムを、外部イベントからの影響が及ぼされて実行が中断する個所を境として、外部イベントの影響が及ぼされることなく内部演算処理に専念できる部分（スレッド）を単位として複数のプログラム片に分割し、複数のスレッドからなるプログラムに変換される。なお、スレッドの分割を、言語処理系で自動的に行うことができる。

【0 0 3 1】スレッドは、当該イベントの到着によって同期を制御しながら起動され、起動されたスレッドは、外部イベントの影響が及ぼされることなく実行される。外部イベントの影響が及ぼされる個所は、予めスレッドの切れ目として特定されているので、イベント処理を、ユーザプログラムとは独立して計算機ハードウェア又は基本ソフトウェアでまとめて実行することができ、ユーザプログラムでは、スレッド実行として内部演算処理に専念することができる。これによって、並列及び分散処理で頻発するイベント処理をユーザプログラムから隠蔽し、かつ、オーバーヘッドを従来に比べて軽減することができる。

【0 0 3 2】したがって、本実施の形態によれば、スレッド実行後に割込み処理をまとめて行うことができるので、外部イベント割込みが内部演算処理に影響を及ぼすことがなくなる。これによって、ユーザプログラムの実行と割込み処理ルーチンとの間のプログラム実行環境の切替を行う回数を著しく低減し、処理のオーバーヘッドを従来よりも軽減することができる。

【0 0 3 3】また、ユーザプログラムでは、内部演算処理をスレッド実行として排他的に実行させることができるので、外部イベントに関連する同期処理の記述を、イベントを契機とするスレッド間の実行依存関係に基づいたプログラムとして構成することができる。これによって、ユーザプログラムでは、イベント間の同期制御を考慮する必要がなく、並列及び分散プログラムの設計及び開発が容易となる。

【図面の簡単な説明】

【図 1】 本発明において実行されるスレッドプログラムの構造を示すプログラムブロック図である。

【図 2】 本発明による多重並行処理装置の全体構成を示すブロック図である。

【図 3】 本発明による多重並行処理方法を実施するハードウェアのブロック図である。

【図 4】 図 3 の外部装置、記憶装置及び演算装置内部のソフトウェアの動作過程を示す図である。

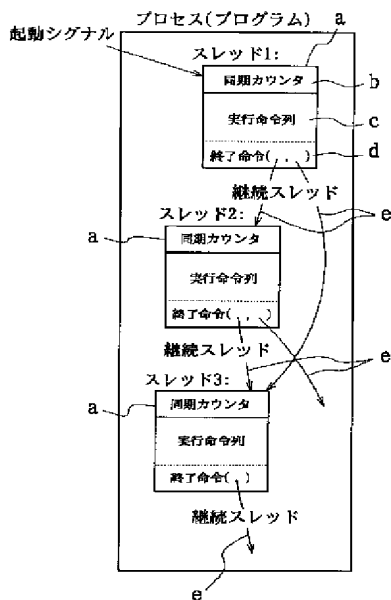
【符号の説明】

- 1 イベント処理装置
- 2 内部演算装置

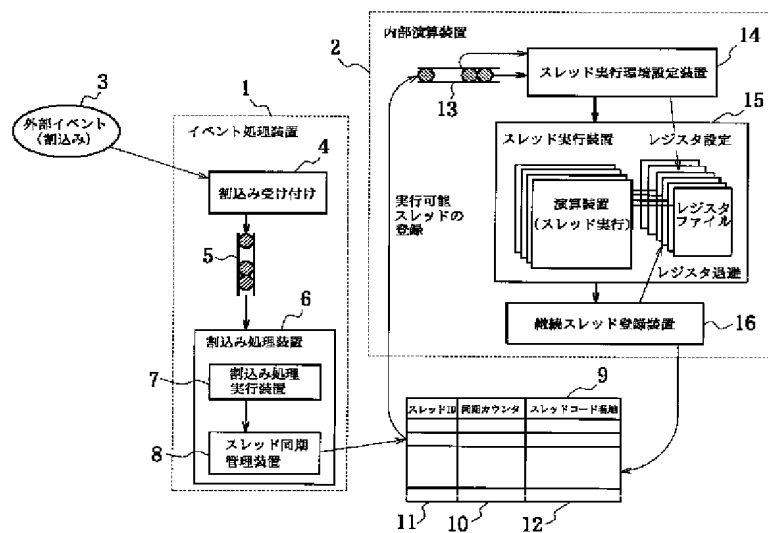
- 3, 27 外部イベント（割込み）  
 4 割込み受付装置  
 5, 28 割込みキュー  
 6 割込み処理装置  
 7 割込み処理実行装置  
 8 スレッド同期管理装置  
 9, 31 同期待ちスレッド管理表  
 10 同期カウンタ  
 11 スレッドID  
 12 スレッドコード番地  
 13, 32 実行待ちスレッドキュー  
 14 スレッド実行環境設定装置  
 15 スレッド実行装置  
 16 継続スレッド登録装置

- 21 ハードウェア装置  
 22 演算装置  
 23 外部装置（割込み装置）  
 24 記憶装置  
 25 バス  
 26 スレッド実行管理部  
 29 スレッド実行部  
 30 割込み処理実行部  
 a スレッドコード  
 10 b スレッド起動条件（同期カウンタ）  
 c スレッド実行命令列  
 d スレッド実行終了命令  
 e 継続スレッド情報（ポインタ）

【図1】



【図2】



【図4】

【図3】

